AFS

ATTORNEY DOCKET NO.: EMC00-03(00011)
LARGE ENTITY

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

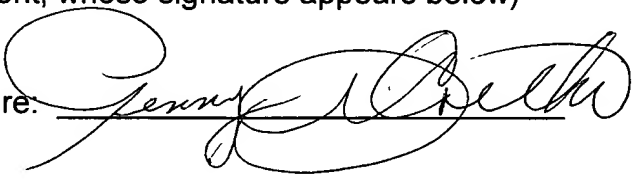
Applicant: Clifford E. Kahn
Serial No.: 09/611,913
For: METHODS AND APPARATUS FOR CONTROLLING ACCESS TO A RESOURCE
Filing Date: July 7, 2000
Examiner: Carl G. Colin
Art Unit: 2136
Conf. No.: 7737

Certificate of Mailing Under 37 C.F.R. §1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: **MAIL STOP APPEAL BRIEF-PATENTS**, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on:

Date: March 3, 2006

By: Penny A. Coelho
(Typed or printed name of person mailing Document, whose signature appears below)

Signature: 

MAIL STOP APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER

Sir:

Enclosed is/are:

[x] Transmittal Letter (this form, 2 pages, in duplicate), Total Pages: 4;
[x] Appeal Brief including Appendix, Total Pages: 66;
[x] Return Receipt Pre-paid Postcard (in duplicate), Total postcards: 2;
[x] Authorization to charge Deposit Account No. 50-3735, if necessary;
[x] Check in the amount of: $500.00.

Applicant hereby petitions for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-3735.

If the enclosed papers or fees are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned collect at (508) 616-9660, in Westborough, Massachusetts.

Respectfully submitted,

Paul P. Kriz, Esq.
Attorney for Applicant
USPTO Registration No.: 45,752
Chapin Intellectual Property Law, LLC
Westborough Office Park
1700 West Park Drive
Westborough, Massachusetts 01581
Telephone: (508) 616-9660
Facsimile: (508) 616-9661

Attorney Docket No.: EMC00-03(00011)

Dated: March 3, 2006

ATTORNEY DOCKET NO.: EMC00-03(00011)
LARGE ENTITY

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Clifford E. Kahn
Serial No.:	09/611,913
For:	METHODS AND APPARATUS FOR CONTROLLING ACCESS TO A RESOURCE
Filing Date:	July 7, 2000
Examiner:	Carl G. Colin
Art Unit:	2136
Conf. No.:	7737

Certificate of Mailing Under 37 C.F.R. §1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: **MAIL STOP APPEAL BRIEF - PATENTS**, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450 on:

Date: March 3, 2006

By: Penny A. Coelho
(Typed or printed name of person mailing Document, whose signature appears below)

Signature: Penny A. Coelho

MAIL STOP APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is being filed within 2 months of the Notice of Appeal filed on January 3, 2006.

03/07/2006 DEMMANU1 00000014 09611913
01 FC:1402 500.00 OP

APEAL BRIEF

(i) Real party in interest.

EMC Corporation, a corporation organized and existing under the laws of the Commonwealth of Massachusetts and having a usual place of business at 35 Parkwood Drive, Hopkinton, Massachusetts 01748.

(ii) Related appeals and interferences.

None

(iii) Status of Claims

Claims 1, 6, 9, 12-14, 19, 24, 27, 30-32, 45, 52-53, 55-82 stand rejected under 35 U.S.C. 102(e) as being unpatentable over Calvignac (U.S. Patent 6,539,394).

(iv) Status of Amendments.

The amendments submitted in response to the final office action mailed September 2, 2005 were not entered because the Examiner felt that the amendments added new matter. Applicant respectfully submits that the Examiner's failure to enter the amendments were improper because Applicant merely attempted to amend claim 1 to include (word for word) the limitations of pending corresponding dependent claim 9. This amendment did not amount to adding new matter to the application. Additionally, Applicant attempted to amend claim 63 to include (word for word) the limitations of pending corresponding dependent claim 65. Both of these amendments were submitted in the response to a Final Office Action for purposes of expediting prosecution of the present application to issuance.

Since the amendments submitted in response to the Final Office Action were not entered, the pending claims submitted in the Appendix to this Appeal Brief reflect the pending claims prior to the mailing of the final office action mailed on September 2, 2005.

(v) Summary of Claimed Subject Matter

The subject application includes 8 independent claims, namely, claims 1, 19, 45, 52, 58, 63, 70, and 76.

In general, the claims at issue in the subject application are directed to access control via rule processing as discussed at page 6 line 22 to page 13 line 14. The Applicant has added underlining to text in the specification to highlight some of the details found in some of the independent claims and corresponding dependent claims. In many cases below, the specification includes additional support for the claim limitations other than that cited by the Applicant.

An access control administrator can create and store rules according to this invention in a master set of rules (350-4, FIG. 5). Upon receipt of an access request (301, FIG. 2) requesting a type of access by a requestor to a resource, the system of the invention processes one or more filter operations that relate to one or more portions of information in the access request (e.g., that relate to the requestor submitting the access request, and/or to a type of access requested, and/or to the resource to which access is requested) to determine which rules from the master set of rules might apply to an access control decision (or a query request) based on the access request. Once the filter operations are complete, a selected set of rules is produced that define rule operations that when processed will grant certain access to certain requestors to the requested resource.

During the creation of rules, the administrator of access control can create rule operations based on conditions defined by relations or the rules may be unconditional or a combination of conditional and unconditional. Once the system of the invention determines the selected set of rules, the system then sequentially processes rule operations from the first rule that applies (based on the filter operations) to the last. If a rule contains a disregard instruction, rule processing can be further altered. A disregard instruction may cause rule processing to complete and not continue on to other rules that were determined to apply based on the filter operations. Alternatively, a disregard instruction can cause rule processing to be further limited to only processing certain rules or selected rule operations within the remainder of the unprocessed selected set of rules. In this manner, a rule set creator can create rules which halt the granting of permissions (i.e., enabling of access) in all circumstances or which limit further rule applications to certain conditions or circumstances. Thus, if a rule grants certain permissions to a certain requestor and includes a disregard instruction, the rule set creator can thus be assured that if the system selects this rule (i.e., is included in the selected set of rules by the processing of the invention) based on filter operation processing, when the system processes the selected rule (i.e., its rule operations), the disregard instruction will either halt all further rule operation processing or will limit such processing to compliance with certain "disregard conditions". If a disregard instruction halts rule processing outright, this enables a rule set creator to position the rule including such a halt disregard instruction ahead or behind other rules in the rule set and if multiple rules appear to apply to an access control decision based on the filter operations, the first rule containing the disregard instruction will cut-off further rule processing.

If the disregard instruction is conditional, then it might cause further rule processing to disregard (i.e., to not process) certain rule operations,

while allowing other remaining rule operations in the selected set of rule (initially selected by the filter operation(s)) to be processed. As an example, a disregard instruction might instruct a rule engine (as will be further explained) that performs rule processing to disregard any further rule operations relating to payroll data. As rule processing continues, the rule engine would thus not process rule operations in other rules within the selected set of rules that have an effect on payroll data.

Relations defining one or more conditions can also be used within rule operations themselves. For example, a relation might precede a disregard instruction, thus allowing conditional rule set processing. For example, if a requestor (e.g., a user requesting access to some data or other resource) were a staff employee, a disregard instruction preceded by a condition might state that if the requestor is in the staff department, then disregard all rules (or rule operations) that relate to granting permissions to payroll data. This conditional disregard instruction might be useful to limit staff employee access to payroll data. Since rule operations (including disregard instructions) can be general in nature, as in these and other examples provided herein, the access control system of the invention allow a security administrator to more generally express a security policy as a set of rules which can be applied system wide. Since the rules are a more concise representation of an access control policy than are a large set of ACLs and/or permissions settings for resources, the invention makes the security administrators job of setting up access control easier and makes verification of the security policy more straightforward.

Using the rule set structure and processing defined by the invention, a rule set creator can essentially program an access control scheme using careful rule creation and positioning of rules within the rule set. Since rules can include disregard instructions, processing may perform less than all rule operations from the selected set of rules. In

some embodiments, this allows an administrator to create a rule hierarchy which can result in processing only a subset of rule operations from the selected set of rules that appear to be applicable to the access control decision based on one or more initially applied filter operations. Since rules can contain nested or deeper conditions than those tested by the filter operations, the system allows an administrator to create rules that can define unique roles, types of access and permissions to precisely control access to a resource. In other words, filter operations can be used to filter out a high level set of rules based on some criteria, while conditions and disregard instructions can be placed within rules themselves to further define and "filter" the granularity of access control (i.e., the granting or denial of permissions to a resource) provided by rule processing. Administrators can thus create tiered access control schemes the first select some rule based on certain factors (via filter operations), and then during application of those rules, can selectively (i.e., conditionally, using relations and disregard instructions) further clarify the granting or denial of permissions to resources for requestors.

More specifically, in one embodiment, the present invention provides a method for providing access control in a computing system environment. The method comprises the steps of receiving an access request and selecting, based on the access request, a selected set of rules containing at least one rule from at least one master set of rules. Since certain rules may be pre-selected in this manner, the system of the invention avoids having to completely fully process all rules in the access control rule set (i.e., the master set of rules). Once the method determines the selected set of rules (either all at once or on a rule by rule basis), the system then performs at least one rule operation in the rule (or rules) in the selected set of rules to produce an access control decision until either one of a rule operation including a disregard instruction is performed to limit performance of rule operations in the selected set of

rules and/or until all rule operations in the selected set of rules that are applicable to the access control decision are performed. Since a disregard instruction can either halt rule processing, or can cause processing to mark or otherwise indicate certain other remaining rule operations to be disregarded in further rule processing, even though those other rule operations in the selected set of rules have not yet been processed, the developer of the rule set can create rules that are assured to only grant certain amounts or levels of permissions or access. Filter operations can be separate from the rules, or may be included as pre-ambles to each rule.

According to another configuration, the step of performing includes the step of producing an access control decision indicating whether to allow access, on behalf of a requestor submitting the access request, to an resource in the computing system environment. The resource and/or the requestor providing the access request may both be local to a resource server (e.g., a computer system, workstation, access control software application, library, etc.) implementing the access control system, or, the access control processing explained herein can be performed on a computing system that is different than the one containing, providing or serving the resource or that supports the requestor. The requestor may be a computer user acting in a specific role, or may be a software application operating on behalf of such a user, or operating in an autonomous or automated manner.

In another configuration, the step of selecting includes the steps of determining an identity of the resource in the computing system environment to which access is requested in the access request and applying at least one filter operation, using the identity of the resource, for rules in the master set of rules to produce the selected set of rules for use in determining the access control decision to the resource. In this

configuration, rule are selected for rule operation processing based on the resource to which access is requested in the access request.

In yet another configuration, the method includes the operation of determining a role identity of a requestor submitting the access request. In this configuration, the step of applying applies the filter operation (one or more), using the role identity of the requestor submitting the access request in combination with the identity of the resource, for rules in the master set of rules to produce the selected set of rules for use in determining the access control decision to the resource. Thus, rule selection can be based on the resource and/or the requestor asking for access to the resource and/or the type of access requested.

The invention also provides configurations in which at least one rule in the selected set of rules contains a rule operation including a unconditional disregard instruction. In such a configuration, the step of performing includes the steps of performing less than all rule operations defined within the rule(s) in the selected set of rules by sequentially performing rule operations in each rule in the selected set of rules until the unconditional disregard instruction is performed thereby terminating (e.g., halting or disabling) the performance of any remaining rule operations in the selected set of rules. As noted above, this allows the rule set creator the ability to create rules that can specifically grant certain permissions and then stop rule processing to be sure only those permissions are granted if that rule is processed, even though other rules may appear to apply to the access control decision based on the initial rule selection process using filter operations.

In another configuration, the selected set of rules is arranged hierarchically such that rules containing rule operations that are more specific are performed before rule operations that are more general. Thus if a rule that is higher up in the hierarchy of a selected set of rules (the all appear to apply to the access control decision) is performed first and

contains a disregard instruction, for instance, this rule will be the last one to be performed. Thus a hierarchical arrangement of rules allows a rule set creator to customize how rule operations are applied in the event more than one rule is selected based on the filter operations.

In other configurations, at least one rule in the selected set of rules contains a rule operation including a disregard instruction including disregard criteria. In these configurations, the operation or step of performing limits performance of rule operations in the selected set of rules by performing the disregard instruction containing disregard criteria such that at least one rule operation in any remaining rule operations in the selected set of rules is disregarded from further performance. Thus, a disregard instruction can include criteria that indicates or identifies other rules or specific rule operations that should not be processed when rule processing continues.

In another configuration, the system of the invention evaluates the disregard criteria against any remaining unperformed rule operations in the selected set of rules and marks any remaining unperformed rule operations in the selected set of rules that match the disregard criteria to be disregarded from further rule processing.

In another configuration, the step of selecting includes the steps of determining an identity of a resource in the computing system environment to which access is requested in the access request and applying at least one filter operation, using the identity of the resource, for rules in the master set(s) of rules to produce the selected set of rules for use in determining the access control decision to the resource. This method further includes the step of determining a role identity of a requestor submitting the access request. Also, the step of performing sequentially processes each rule operation in the selected set of rules using the role identity of the requestor submitting the access request in combination with the identity of the resource to determine if the requestor

using the role identity can access the resource. Thus, filter operations can test resource identities, requestor identities and even types of access requested to determine if a rule appears to apply to the access control decision.

In another configuration, the disregard instruction is a conditional instruction that has a condition, such as, for example, being based on the role identity of the requestor submitting the access request, that must be met before the disregard instruction is performed. Thus, the system can determine who is requesting access immediately before performing a disregard instruction, and if a certain identity is requesting, the rule operation processing might be disregarded or limited to processing only certain rules or rule operations thereafter, whereas if another requestor (i.e., having a different identity) is requesting access, then rule operations may continue without alteration of processing (i.e., all rule operations are processed as normal).

In another configuration, at least one rule in the selected set of rules contains a relation that defines a condition based on a group definition. In this configuration, at least one of the steps of selecting and performing includes the step of performing the relation to determine if either a requestor, an access, and/or a resource specified in the access request satisfy the condition based on the group definition. Thus relations can be used within filter operations and rule operations to perform conditional tests using, for example, requestor, access and resource information obtained from the access request.

Support for each claim limitation can be found at a respective location in the specification as identified in the following claims. Additional areas of support in the specification for each claim limitation have not been cited to avoid unnecessary clutter.

1. A method for providing access control in a computing system (100) environment, the method comprising the steps of:

receiving an access request (301, FIG. 2) (step 400, FIG. 6, page 42 line 7 to page 43 line 7);

selecting, based on the access request (301), a set of rules containing at least one rule from a master set of rules (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7), and

producing an access control decision (405, FIG. 6, page 42 line 7 to page 43 line 7) based on performing rule operations in a given rule of the selected set of rules (step 402, FIG. 6, page 42 line 7 to page 43 line 7) by sequentially performing rule operations in the given rule until performing a disregard instruction (step 403, FIG. 6, page 42 line 7 to page 43 line 7), the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing (page 27 line 26 to page 28 line 5, page 38 line 1 to page 39 line 22); and

after performing the disregard instruction (step 404, FIG. 6, page 42 line 7 to page 43 line 7) in the given rule:

i) evaluating the disregard criteria against any remaining unperformed rule operations in other rules of the selected set of rules, the other rules being rules other than the given rule (page 44, lines 7-14);

ii) marking any remaining unperformed rule operations in the other rules of the selected set of rules that match the disregard criteria to be disregarded from further rule processing (page 44, lines 7-14); and

iii) executing remaining unmarked rule operations in the other rules in the selected set of rules (page 44, lines 7-14).

19. A computer system (200, FIG. 2, page 25 line 3 to page 28 line 6) configured to provide access control, the computer system comprising:

at least one input/output interface (210, FIG. 2, page 25 line 3 to page 28 line 6);

a processor (220, FIG. 2, page 25 line 3 to page 28 line 6);

a memory system (208, FIG. 2, page 25 line 3 to page 28 line 6) encoded with an authorization program;

at least one authorization database (350, FIG. 2, page 25 line 3 to page 28 line 6);

an interconnection mechanism (page 13 line 15-27) coupling the processor, the at least one input/output interface, the memory system, and the at least one authorization database (page 13 line 15-27);

based at least in part on the processor executing the authorization program, the processor supporting steps of:

receiving an access request (301, FIG. 2) (step 400, FIG. 6, page 42 line 7 to page 43 line 7);

selecting, based on the access request, a set of rules containing at least one rule from a master set of rules (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7);

producing an access control decision (405, FIG. 6, page 42 line 7 to page 43 line 7) based on performing rule operations in a given rule of the selected set of rules by sequentially performing rule operations in the given rule until performing a disregard instruction (step 403, FIG. 6, page 42 line 7 to page 43 line 7), the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing (page 27 line 26 to page 28 line 5, page 38 line 1 to page 39 line 22); and

after performing the unconditional disregard instruction (step 404, FIG. 6, page 42 line 7 to page 43 line 7) in the given rule:

i) evaluating the disregard criteria against any remaining unperformed rule operations in other rules of the selected set of rules, the other rules being rules other than the given rule (page 44, lines 7-14);

ii) marking any remaining unperformed rule operations in the other rules of the selected set of rules that match the disregard criteria to be disregarded from further rule processing (page 44, lines 7-14); and

iii) executing remaining unmarked rule operations in the other rules in the selected set of rules (page 44, lines 7-14).

45. A method for controlling applicability of rule operations in a rule-based access control system (200, FIG. 2, page 25 line 3 to page 28 line 6), the method comprising the step of:

selecting at least two rules for performance to determine an access control decision, the at least two rules including a first rule and a second rule (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7);

performing a rule operation in the first rule of the at least two rules, the rule operation including a disregard instruction that, when performed, causes non-performance of at least one rule operation in the second rule that is disregarded based on the disregard instruction (step 403, FIG. 6, page 42 line 7 to page 43 line 7); and

performing at least one rule operation in the second rule other than the at least one rule operation in the second rule that is disregarded (page 7 line 3 to page 9 line 5) (FIGS. 5 and 6, page 43 line 25 to page 45 line 17).

52. A method for providing access control in a computing system environment (200, FIG. 2, page 25 line 3 to page 28 line 6), the method comprising the steps of:

receiving an access request (301, FIG. 2) (step 400, FIG. 6, page 42 line 7 to page 43 line 7);

selecting, based on the access request, a set of rules containing multiple rules from at least one master set of rules, at least one of the multiple rules including multiple rule operations to be performed in sequential order (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7);

for a first rule of the multiple rules:

performing a filter operation associated with the first rule to identify whether to execute any rule operations in the first rule (FIG. 5, page 33 line 19 to page 38 line 15); and

performing multiple operations in the first rule to determine whether to provide access to a storage system in response to receiving the access request (step 403, FIG. 6, page 42 line 7 to page 43 line 7), the first rule including a disregard instruction that, when executed, limits performance to fewer than all rule operations in a second rule of the selected set of rules as specified by disregard criteria in the disregard instruction (page 7 line 3 to page 9 line 5) (FIGS. 5 and 6, page 43 line 25 to page 45 line 17) (step 403, FIG. 6, page 42 line 7 to page 43 line 7).

58. A method for providing access control in a computing system environment (200, FIG. 2, page 25 line 3 to page 28 line 6), the method comprising:

receiving an access request (301, FIG. 2) (step 400, FIG. 6, page 42 line 7 to page 43 line 7);

in response to receiving the access request, selecting a set of rules for processing to determine whether to permit the access request (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7);

during processing of the set of rules, performing a conditional disregard rule operation in the set of rules (page 7 line 3 to page 9 line 5) (FIGS. 5 and 6, page 43 line 25 to page 45 line 17);

based on performing the conditional disregard rule operation, disregarding execution of at least one rule operation other than the conditional disregard rule operation in the set of rules as specified by the conditional disregard rule operation (step 403, FIG. 6, page 42 line 7 to page 43 line 7); and

after performing the conditional disregard rule operation, performing at least one other rule operation in the set of rules not specified by disregard criteria in the conditional disregard rule operation (page 7 line 3 to page 9 line 5) (FIGS. 5 and 6, page 43 line 25 to page 45 line 17).

63. A method for providing access control in a computing system environment (200, FIG. 2, page 25 line 3 to page 28 line 6), the method comprising:

receiving an access request (301, FIG. 2) (step 400, FIG. 6, page 42 line 7 to page 43 line 7);

in response to receiving the access request, selecting a first set of rules and a second set of rules for processing to determine whether to permit the access request, the first set of rules and the second set of rules each including multiple rule operations (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7);

during processing of the first set of rules, performing a disregard rule operation in the first set of rules (step 403, FIG. 6, page 42 line 7 to page 43 line 7); and

based on performing the disregard rule operation, disregarding execution of at least one rule operation in the second set of rules as identified by the disregard rule operation (page 7 line 3 to page 9 line 5) (FIGS. 5 and 6, page 43 line 25 to page 45 line 17).

70. A method for providing access control in a computing system environment (200, FIG. 2, page 25 line 3 to page 28 line 6), the method comprising:

receiving an access request to access data in the computing system environment (301, FIG. 2) (step 400, FIG. 6, page 42 line 7 to page 43 line 7);

comparing the access request to a master rule set, each rule in the master rule set including a filter and a corresponding set of rule operations to be performed pending evaluation of the filter condition (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7); and

for each rule containing a filter operation that evaluates to indicate execution of rule operations of that rule, executing the rule operations of that rule (FIG. 5, page 33 line 19 to page 38 line 15);

during execution of rule operations of that rule (step 403, FIG. 6, page 42 line 7 to page 43 line 7), executing a first conditional disregard instruction that establishes a first set of pre-conditions that must be met in successive rules in the master rule set in order for those successive rules to be executed after the rule containing the first conditional disregard instruction has been executed (step 403, FIG. 6, page 42 line 7 to page 43 line 7); and

executing at least one successive rule in the master rule set for which the access request meets the filters of those successive rules, and for which the first set of pre-conditions established by executing the first conditional disregard instruction are also met (page 7 line 3 to page 9 line 5) (FIGS. 5 and 6, page 43 line 25 to page 45 line 17).

76. A computer program product (200, FIG. 2, page 25 line 3 to page 28 line 6) having a computer-readable medium including computer program logic encoded thereon that when executed on a computer system provides a method for controlling access to a resource, and wherein when the computer program logic is executed on a processor in the computer system, the computer program logic causes the processor to perform the operations of:

receiving an access request to access data in the computing system environment (301, FIG. 2) (step 400, FIG. 6, page 42 line 7 to page 43 line 7);

comparing the access request to a master rule set, each rule in the master rule set including a filter and a corresponding set of rule operations to be performed pending evaluation of the filter condition (350-4, FIG. 5) (step 401; FIG. 6, page 42 line 7 to page 43 line 7); and

for each rule containing a filter operation that evaluates to indicate execution of rule operations of that rule, executing the rule operations of that rule (step 403, FIG. 6, page 42 line 7 to page 43 line 7);

during execution of rule operations of that rule, executing a first conditional disregard instruction that establishes a first set of pre-conditions that must be met in successive rules in the master rule set in order for those successive rules to be executed after the rule containing the first conditional disregard instruction has been executed (step 403, FIG. 6, page 42 line 7 to page 43 line 7); and

executing at least one successive rule in the master rule set for which the access request meets the filters of those successive rules (FIG. 5, page 33 line 19 to page 38 line 15), and for which the first set of pre-conditions established by executing the first conditional disregard instruction are also met (page 7 line 3 to page 9 line 5) (FIGS. 5 and 6, page 43 line 25 to page 45 line 17).

Support for claim 57 can be found in figure 5 and at corresponding text at page 33 line 19 to page 42 line 6 in the subject application as well as elsewhere throughout the specification.

Support for claim 60 can be found in figure 5 and at corresponding text at page 33 line 19 to page 42 line 6 in the subject application as well as elsewhere throughout the specification.

Support for claim 62 can be found in figure 6 and at corresponding text at page 30 line 29 to page 42 line 6 in the subject application as well as elsewhere throughout the specification.

Support for claim 65 can be found in figure 5 and at corresponding text at page 33 line 19 to page 42 line 6 in the subject application as well as elsewhere throughout the specification.

Support for claim 67 can be found in figure 5 and at corresponding text at page 33 line 19 to page 42 line 6 in the subject application as well as elsewhere throughout the specification.

Support for claim 72 can be found in figure 5 and at corresponding text at page 33 line 19 to page 42 line 6 in the subject application as well as elsewhere throughout the specification.

(vii) Argument

Note that the Applicants attempt to set forth the Examiner's argument to the best of their ability.

Rejection of Claims 1 and 19

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claims 1 and 19. Applicant respectfully traverses the Examiner's rejections on grounds that the cited reference does not teach every claim limitation.

As will be further discussed below, Calvignac supports a two-stage process of first creating subsets of rules and thereafter performing a matching process to select a set of rules. The rules in Calvignac are applied to make a forwarding decision of a packet. There is no indication in Calvignac executing a disregard rule operation that causes non-execution of certain other rule operations as indicated by the disregard rule operation. In fact, Calvignac does not mention that rules in any of the subsets reference each other in any way whatsoever. The claimed invention indicates that a disregard rule operation can reference other rule operations that are to be disregarded. The teachings of Calvignac provide no such support.

For example, the cited reference discloses a technique of building subsets of rules and applying one or more selected subset of rules to presumably make a forwarding decision. In one embodiment of Calvignac, a decision tree is used to select which of multiple rules to apply in order to make a forwarding decision of a data packet. As will be further discussed below, Applicant respectfully submits that the technique of building subsets of rules (as cited by the Examiner) and creating a decision tree in Calvignac (which is used prior to performing the rules to make a data packet forwarding decision) is not equivalent to the claimed technique of sequentially performing rule operations in a selected set of rules, especially one in which execution of a disregard operation in a given rule affects execution of other rule operations in the given rule.

More specifically, the Examiner cites broad portions of Calvignac at column 6, line 15 to column 8, line 47 (FIGS. 3A-3D) because the Examiner

cannot specifically point out how Calvignac anticipates the claimed invention. With emphasis added by the Applicant to indicate building subsets of rules, the cited section in Calvignac reads as follows:

> FIG. 3A depicts a more detailed flow chart of one embodiment of a method 110 in accordance with the present invention. The method 110 is one embodiment of the method 100 and is used in a similar manner to reduce the amount of testing required. The minimum and maximum values of the ranges for the filter rules are used to determine subsets of rules, via step 112. Each subset is for a different dimension for which the filter rules utilize a range of values. For example, if the IP header is used as a key, the ranges can be in one or more of five dimensions, one for each field in the IP header. Thus, there is a subset in one or more of the five dimensions. Furthermore, the subsets are distinct. Thus, each subset includes a different portion of the filter rules undergoing testing. The subsets include rules which do not intersect. Thus, in a particular dimension, the subset of rules corresponding to that dimension do not intersect in that dimension. The minimum and maximum values of the filter rules are used to ensure that the filter rules in a particular subset do not intersect.
>
> Once the subsets of filter rules are obtained, the key undergoing test is tested against each of the subsets to determine which, if any, of the rules from each subset the key may match, via step 114. Preferably, only the field of the key corresponding to the appropriate dimension is tested against the subset for that dimension. For example, if a subset of filter rules which do not intersect in the source address dimension is obtained, the source address of the key is tested against the subset of filter rules for the source address. The minimum and maximum values of the subset are used to test a key in a particular dimension. If the key matches one of the rules in a particular subset, that key may match that rule. In a preferred embodiment, a maximum of one rule may be a match for the key in a particular dimension because each subset preferably includes only those

filter rules which do not intersect in the particular dimension being tested. Thus, using step 114, the number of filter rules which the key may match may be greatly reduced to a number of rules no larger than the number of dimensions for which the filter rules use ranges of values.

Once the rules which the key may match are isolated, the key is explicitly tested against these rules, via step 116. Step 116 thus includes testing each dimension for each of the rules. If the key matches more than one rule, then the priority of the rules is determined and the rule with the highest priority controls. Thus, the number of rules against which each field of the key must be explicitly tested is greatly reduced. Consequently, the application of filter rules is made simpler and more efficient.

FIG. 3B depicts a more detailed flow chart of a method for performing step 112, determining the subsets. A sweep is performed in each dimension for which the plurality of rules has a range to determine a subset in each dimension, via step 122. Using the minimum and maximum values for each dimension, each sweep determines a distinct subset of non-intersecting rules. Thus, the sweep process is then repeated for the remaining dimensions: the destination address, the source port, the destination port, and the protocol. Consequently, a subset of non-intersecting rules is obtained for each dimension using step 122.

A subset of the subsets determined in step 122 is then selected as the first subset, via step 124. The largest subset, the subset including the most filter rules, is preferably selected in step 124 as the first subset. The filter rules contained in the first subset and the corresponding dimension are then temporarily discarded, via step 126. In other words, the filter rules in the first subset and corresponding dimension will not be considered when forming further subsets in step 112.

Using the remaining filter rules and dimensions, subsets of non-intersecting rules are determined for each of the dimensions, via step 128. Step 128 preferably utilizes sweeps which are similar to those discussed above with respect to step 122. Thus, step 128 uses the minimum and maximum values of the filter rules to provide subsets of rules which are nonintersecting. A next subset is selected from the subsets determined in step 128 and both the filter rules in that subset and the corresponding dimension are discarded, via step 130. In a preferred embodiment, step 130 selects the largest subset of the subsets formed in step 128 as the next subset. Steps 128 and 130 are then repeated for the remaining dimensions, via step 132. Thus, via step 128 through 132, subsets of non-intersecting rules are built for the remaining dimensions. In a preferred embodiment, all dimensions can have a subset of non-intersecting rules. However, in the case where this is not possible, preferably only the last dimension may have intersecting rules.

Referring back to FIG. 3A, once these subsets are built, the key can be tested by testing one field, or dimension, of the key against each subset using step 114 of the method 110. As a result a relatively small number of rules which the key may match is obtained. Because only one field of the key is tested in each dimension, testing is relatively simple. This testing can greatly narrow the number of filter rules which a key may match. In a preferred embodiment, the maximum number of rules which a key can match is equal to the number of dimensions. Using step 116, all fields of the key may then be explicitly tested against the filter rules obtained in step 114. Thus, the filter rules which the key may match can be relatively rapidly and easily determined.

FIGS. 3C-3D depict a method for performing a sweep in step 122. Using the minimum and maximum values for each dimension, each sweep determines a distinct subset of non-intersecting rules. The sweep is commenced at the minimum value for the dimension, via step 135. It is determined whether the minimum value for a filter rule has been encountered, via step 136. If not, the

sweep is continued, via step 137, until a minimum value is encountered. Once a minimum value is encountered, it is determined whether more than one filter rule shares the same minimum value, via step 138. If not, then the filter rule encountered is selected as part of the subset for the dimension, via step 139. If more than one filter rule shares the same minimum value, then one filter rule is selected as being part of the subset and the other filter rule(s) sharing the minimum value are discarded for the dimension, via step 140. In a preferred embodiment, the filter rule having the lowest index value is selected as part of the subset in step 140. The sweep is then continued, via step 141.

It is determined whether the minimum value for another filter rule is encountered before the maximum value of the selected filter rule has been encountered, via step 142. If the minimum value for another filter rule is encountered, then the filter rule is discarded, via step 143. It is determined whether the maximum value for the selected filter rule has been encountered in the sweep, via step 144. If no, then the sweep is continued, via step 145. This process of discarding filter rules is continued until the maximum value of the selected filter rule is encountered. Thus, filter rules which have a range that overlaps the selected rule are discarded for the dimension of interest. Once the maximum value for the selected rule is encountered, the sweep is continued, via step 137. Thus, the sweep will continue until another minimum value for another filter rule is encountered. This process of selecting filter rules and discarding other filter rules having overlapping ranges is continued until the rules are all either selected or discarded or until the end of the dimension is reached. Thus, a non-interesecting set of filter rules for a dimension can be obtained. The non-intersecting filter rules in the subset are also ordered from smallest to largest minimum values. The method 122 can be repeated for other dimensions.

For example, assume that the above five fields of the IP header of a packet are used as a key. Also assume that the plurality of filter rules utilize ranges for the

source address, destination address, source port, destination port and protocol. A sweep may be performed first for the source address. The sweep commences at the smallest possible value of the source address, zero, using step 135. When the sweep reaches the smallest minimum value for a rule, that rule is selected as part of the subset using step 139 or, if two or more rules have the same minimum value one of the filter rules, using step 140. Any filter rule which intersects the selected filter rule is discarded using step 140 or 142. The discarded filter rules are not part of the subset. Thus, any filter rule having a range which overlaps the range of the selected filter rule is not part of the subset. After reaching the maximum value of the selected rule, the sweep continues until a next minimum value is reached. This process is continued until a subset of non-intersecting filter rules is obtained for the source address. Furthermore, the sweep ensures that the subset includes rules which are ordered from smallest to largest minimum values. The sweep process is then repeated for the remaining dimensions: the destination address, the source port, the destination port, and the protocol. (emphasis added)

As shown in the above cited passages, these underlined portion of Calvignac disclose a method of creating or building subsets of filter rules. Applicant disagrees with the Examiner assertion that creation of subsets of filter rules is equivalent to limitations in the claimed invention.

In particular, at column 8 lines 15-19, Calvignac characterizes the process disclosed in Figures 3C and 3D and corresponding text. The Examiner relies on this and other related text to reject the claimed invention. Calvignac specifically states "This process of selecting filter rules and discarding other filter rules having overlapping ranges is continued until the rules are all either selected or discarded or until the end of the dimension is reached." In other words, Calvignac characterizes the process in FIGS. 3A-3D as a process of selecting filter rules to build subsets of rules, not performing rule operations. The Examiner mistakenly interprets the process of selecting rules to mean

performance of rules. If the rules were being performed at this stage in Calvignac as purported bythe Examiner, then the discussion of applying subsets of rules in Calvignac at Figures 3A and 3B would not make sense. The Examiner therefore misconstrues these teachings of Calvignac to reject the claimed invention. Again, the claimed invention entails performing rule operations such as a disregard instruction to identify other types of rule operation to disregard from execution. This means that a rule operation (e.g., the disregard instruction) makes reference to other types of rule operations. Applicant respectfully submits that Calvignac provides no teaching whatsoever that one rule (or rule operation) references or identifies another type of rule (or rule operation), especially in which the rule indicates which other rules to disregard from execution.

Further, note that Calvignac states at column 7, lines 33-35: "Thus, using steps 122 through 132, subsets preferably including non-intersecting rules are built for each of the dimensions."

Applicant respectfully submits that this process does not teach or suggest the claimed invention of: "producing an access control decision based on performing rule operations in a given rule of the selected set of rules by sequentially performing rule operations in the given rule until performing a disregard instruction, the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing."

First, as discussed above, Applicant again points out that the cited passages in Calvignac discloses building subsets of non-intersecting rules based on a sweep technique as recited at column 7, lines 50-53: "Using the minimum and maximum values for each dimension, each sweep determines a distinct subset of non-intersecting rules." Building subsets of non-intersecting rules as done in Calvignac is not equivalent to producing an access control decision by

done in Calvignac is not equivalent to producing an access control decision by sequentially performing rule operations as in the claimed invention. These are two distinct processes.

Also, the claimed invention recites "performing rule operations in a given rule." There is no indication in this portion of Calvignac that a respective rule includes multiple rule operations. Perhaps more importantly, Calvignac does not disclose that a rule operation can be a disregard instruction that, when performed, causes other rule operations in the given rule not to be performed. As discussed above and as will be further discussed below, the rules in Calvignac are not yet applied (or executed with respect) to a key until after detecting which subsets of rules match the key.

More specifically, as discussed above, figures 3B through 3C discuss a technique of building the subsets of rules. Figure 3C and 3D are basically step 122 in figure 3B of building the subsets of rules. Figure 3B illustrates details associated with step 112 in figure 3A. Thus the first steps of figure 3A entail building the subsets of rules as discussed above. Other following steps in figure 3A, namely steps 114 and 116 and corresponding text in Calvignac, illustrate how to use the subsets of rules created in figures 3B through 3D. In other words, Calvignac discloses use of the "built" subsets of rules in FIG. 3A at steps 114 and 116 and corresponding text. Text in Calvignac at column 6 lines 15-63 associated with figure 3A reads as follows:

> FIG. 3A depicts a more detailed flow chart of one embodiment of a method 110 in accordance with the present invention. The method 110 is one embodiment of the method 100 and is used in a similar manner to reduce the amount of testing required. The minimum and maximum values of the ranges for the filter rules are used to determine subsets of rules, via step 112. Each subset is for a different dimension for which the filter rules utilize a range of values. For example, if the

IP header is used as a key, the ranges can be in one or more of five dimensions, one for each field in the IP header. Thus, there is a subset in one or more of the five dimensions. Furthermore, the subsets are distinct. Thus, each subset includes a different portion of the filter rules undergoing testing. The subsets include rules which do not intersect. Thus, in a particular dimension, the subset of rules corresponding to that dimension do not intersect in that dimension. The minimum and maximum values of the filter rules are used to ensure that the filter rules in a particular subset do not intersect.

Once the subsets of filter rules are obtained, the key undergoing test is tested against each of the subsets to determine which, if any, of the rules from each subset the key may match, via step 114. Preferably, only the field of the key corresponding to the appropriate dimension is tested against the subset for that dimension. For example, if a subset of filter rules which do not intersect in the source address dimension is obtained, the source address of the key is tested against the subset of filter rules for the source address. The minimum and maximum values of the subset are used to test a key in a particular dimension. If the key matches one of the rules in a particular subset, that key may match that rule. In a preferred embodiment, a maximum of one rule may be a match for the key in a particular dimension because each subset preferably includes only those filter rules which do not intersect in the particular dimension being tested. Thus, using step 114, the number of filter rules which the key may match may be greatly reduced to a number of rules no larger than the number of dimensions for which the filter rules use ranges of values.

Once the rules which the key may match are isolated, the key is explicitly tested against these rules, via step 116. Step 116 thus includes testing each dimension for each of the rules. If the key matches more than one rule, then the priority of the rules is determined and the rule with the highest priority controls. Thus, the number of rules against which each field of the key must be explicitly tested is

greatly reduced. Consequently, the application of filter rules is made simpler and more efficient. (Emphasis Added)

The Examiner seems to be arguing (at least in part) that the above passage in Calvignac also discloses the claim limitation of: "producing an access control decision based on performing rule operations in a given rule of the selected set of rules by sequentially performing rule operations in the given rule until performing a disregard instruction, the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing." Applicant respectfully disagrees with his assertion.

For example, as discussed above in Calvignac, "once the rules which the key may match are isolated, the key is explicitly tested against these rules" (Calvignac column 6, lines 34-37). There is no indication whatsoever that the process in Calvignac includes performing multiple rule operations for a given rule. In fact, there is no indication that any of the rules in Calvignac include multiple rules. Nor is there an indication that the rule operations are performed in a sequential manner. Perhaps most importantly, there is no indication in Calvignac of sequentially performing rule operations in a given rule until performing a disregard instruction that indicates other rule operations to disregard from performing. If anything, Calvignac discloses execution or application of all rules in a respective subset to a particular key. There is no indication that only a portion of rules in a respective subset of rules are applied to the key. There especially is no teaching or suggestion regarding how execution of one type of rule affects another. Accordingly, Applicant respectfully submits that the rejection under 102 is improper.

During prosecution of the application, the Examiner changes his argument about what portions of Calvignac anticipate the above claim limitation. For

example, the Examiner seems to additionally cite figure 5, column 3, lines 49-53, and column 10, lines 1-14 of Calvignac to reject the claimed invention. The Examiner contends that Calvignac discloses a decision tree, which is used to isolate a portion of a plurality of rules on a leaf path having at least one node.

Applicant respectfully submits that the decision tree in figure 5 of Calvignac also does not anticipate the claimed step of: "producing an access control decision based on performing rule operations in a given rule of the selected set of rules by sequentially performing rule operations in the given rule until performing a disregard_instruction, the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing."

Applicant would first like to explain the additional teachings of Calvignac as cited by the Examiner, which may be best understood at a high level via the illustration in FIG. 4. As shown, and as further discussed in the respective Abstract, a portion of Calvignac is directed to creating a decision tree (Calvignac, FIG. 5) for filtering rules. Nodes in the decision tree enable a computer filter process to reduce a number of filter rules that should be applied to a received data packet for purposes of forwarding the data packet in a network environment (Calvignac, FIG. 1). More specifically, a key in a respective data packet is compared to nodes in the decision tree to determine which of multiple rules to apply to the data packet for forwarding purposes. As shown in FIG. 5 of Calvignac, starting at the top of the decision tree and working downward, rules at respective leafs of the tree are applied to the data packet for forwarding purposes. Each field of a respective key (of a data packet) is viewed as a dimension (Calvignac, col 5, lines 36-54). A key can include multiple dimensions. Thus, when applying the decision tree in Calvignac, the respective nodes of the decision tree are applied to the different fields of the data packet to identify which of multiple final rules will possibly apply to the data packet for

forwarding purposes. In other words, after reducing a set of rules down to a subset of rules at leaf nodes, Calvignac discloses that the key can be explicitly tested against the selected set of rules to determine whether a match exists and if the rule applies to the data packet.

Claim 1 includes limitations not found in Calvignac. For example, claim 1 recites a process of selecting a set of rules and further limiting execution of the selected set of rules during the execution process depending on an execution of a disregard instruction in the executed set of selected rules. That is, claim 1 recites "selecting, based on the access request, a set of rules containing at least one rule from a master set of rules" as well as "producing an access control decision based on performing rule operations in a given rule of the selected set of rules by sequentially performing rule operations in the given rule until performing a disregard instruction, the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing." Further, claim 1 recites: "after performing the disregard instruction in the given rule: i) evaluating the disregard criteria against any remaining unperformed rule operations in other rules of the selected set of rules, the other rules being rules other than the given rule."

In addition to the above arguments, Applicant submits that claim 1 is novel and includes limitations not taught or suggested by the cited art. For example, the claimed invention includes executing a disregard instruction which itself includes disregard criteria identifying a type of rules in the selected set of rules that shall be disregarded from performing. The disregard rule includes disregard criteria for disregarding performance of certain specified types (as indicated by the disregard instruction) of rule operations in the selected set of rules. Thus, performance of a disregard instruction in one rule of the selected set of rules affects future performance of specifically identified other rules in a selected set of rules. The instruction of the "given rule" is performed before further limiting

execution of other rules at run-time when the rules are being applied to make a determination of whether to allow or deny the access request. This is not shown in the cited reference.

As mentioned, the cited references disclose a method of selecting a set of rules and determining whether the rules shall be applied or not applied. There is no mention of executing or performing a rule in the selected set of rules and, based on executing one of the rules, utilizing disregard criteria in a performed rule to identify which other specific rule operations in the selected set of rules to disregard. That is, Calvignac more specifically discloses a two-stage process. A first selection stage involves picking ranges that do not intersect or overlap and creating corresponding subsets of rules. The second stage involves comparing a key in a header of an IP packet to the ranges from the first selection stage of respective rules that have non-overlapping ranges. Application of the rules at the second selection stage determines whether or not to forward a packet. An advantage of the method in Calvignac is to more quickly select which rules shall be applied to a given IP packet prior to actual execution or performance of the rules on the key of the IP packet. Thus, Calvignac solves a different problem than that of the claimed invention. For example, in comparison, the claimed invention enables execution of rules at run-time to prevent other rules from being performed because the rules themselves can include a disregard instruction that limits execution of other rule operations. Calvignac does not provide such capability and therefore does not anticipate the claimed invention.

Claim 1 includes a description of how to carry out a disregard instruction also not disclosed by Calvignac. For example, claim 1 further recites:

"after performing the disregard instruction in the given rule:

i) evaluating the disregard criteria against any remaining unperformed rule operations in other rules of the selected set of rules, the other rules being rules other than the given rule;

ii) marking any remaining unperformed rule operations in the other rules of the selected set of rules that match the disregard criteria to be disregarded from further rule processing; and

iii) executing remaining unmarked rule operations in the other rules in the selected set of rules."

Thus, another difference between Calvignac and the claimed invention is the way in which rules are disregarded. Calvignac does not recite this step because of the nature of "pre-compiling" (e.g., creating subsets of rules) and thereafter applying the so-called matching subsets of rules to identify which of multiple rules shall be applied to make a respective data packet forwarding decision. Again, the claimed invention involves sequentially performing the rule operations to carry out a respective disregard instruction. Performance of the rules and a disregard instruction in a given rule during a sequential execution of the rules dictates which further rule operation in the selected set of rules shall be performed. Calvignac makes no mention that only a portion of a subset of rules that matches a respective key will be applied to the key. For example, Calvignac specifically states that "Once the subsets of filter rules are obtained, the key undergoing test is tested against each of the subsets to determine which, if any, of the rules from each subset the key may match." How does this process in Calvignac teach steps of evaluating, marking, and executing as recited in the claimed invention? There is no indication that one of the rules in an applied subset of rules is a disregard instruction that is evaluated for purposes of marking other rules for non-execution and executing any remaining unmarked rule operations.

Based on these distinctions, Applicant respectfully submits that the rejections of claims 1 and 19 under 35 U.S.C. § 102(e) are improper.

Rejection of Claim 45

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 45. Applicants respectfully traverse the rejection because claim 45 includes limitations not found in any of the cited references. The Examiner fails to notice that claim 45 includes different limitations than claims 1 and 19, but uses the same arguments associated with the rejection of claims 1 and 19 to reject claim 45 as well.

In its entirety, claim 45 reads as follows:

45. A method for controlling applicability of rule operations in a rule-based access control system, the method comprising the step of:

selecting at least two rules for performance to determine an access control decision, the at least two rules including a first rule and a second rule;

performing a rule operation in the first rule of the at least two rules, the rule operation including a disregard instruction that, when performed, causes non-performance of at least one rule operation in the second rule that is disregarded based on the disregard instruction; and

performing at least one rule operation in the second rule other than the at least one rule operation in the second rule that is disregarded.

Applicant submits that claim 45 includes distinctions over the cited prior art and claim 1. For example, claim 45 specifically recites "selecting at least two rules for performance to determine an access control decision, the at least two rules including a first rule and a second rule; performing a rule operation in the first rule of the at least two rules, the rule operation including a disregard

instruction that, when performed, causes non-performance of at least one rule operation in the second rule that is disregarded based on the disregard instruction; and performing at least one rule operation in the second rule other than the at least one rule operation in the second rule that is disregarded."

As discussed above, the cited portions of Calvignac used to reject the claimed invention disclose a two-stage process. A first stage of the two-stage process involves building subsets of rules (e.g., see figures 3A –3D and corresponding text as cited by the Examiner). A second stage of the two-stage process involves testing a key against each subset to identify which rules to apply for purposes of generating a final forwarding decision.

Although Calvignac does not get into any relevant details regarding execution of the rules in the "built" subsets (or at the leaf nodes of the tree as discussed in other embodiments), Calvignac does indicate that rules can be used to make a forwarding decision. However, there is no indication whatsoever in Calvignac to limit application of a number of rules based on performance (e.g., execution) of a disregard rule operation in a given rule that limits operation of other rules. Stated differently, there is no indication in Calvignac that a rule or rule operation itself has an affect on execution of other rules or rule operations, especially one describing a process of "performing a rule operation in the first rule of the at least two rules, the rule operation including a disregard instruction that, when performed, causes non-performance of at least one rule operation in the second rule that is disregarded based on the disregard instruction; and performing at least one rule operation in the second rule other than the at least one rule operation in the second rule that is disregarded."

The cited reference as discussed above in Calvignac indicates that "once the rules which the key may match are isolated, the key is explicitly tested against these rules." There is no indication that the key is tested against only a

portion of a subset of rules. Thus, Calvignac does not teach or even suggest performing a disregard instruction in a first selected rule, performance of the disregard instruction causing non-performance of a rule operation in a second rule, while at least one rule operation in the second rule is still performed. In other words, according to the claimed invention, the second rule includes some non-performed rule operations (e.g., disregarded rule operations based on execution of a disregard instruction in a first rule) and some performed rule operations. For example, there is no indication in Calvignac that only a portion of a given rule (rather than all of a given rule) is ever performed as in the present invention of claim 45. Calvignac only shows how to create and use subsets of rules or a decision tree to select and apply rules that are used to make a forwarding decision.

Also, the Examiner assumes that a rule in Calvignac includes multiple rule operations without providing any basis. Applicant disagrees with this assertion because there is no indication in any of the cited passages that a rule includes multiple rule operations. Perhaps more importantly, Calvignac does not show performing a disregard instruction in one rule, which results in performance of some rule operations and non-performance of other rule operations in a second rule as in the claimed invention. For example, claim 45 recites that at least one rule operation (e.g., in the second rule) is disregarded and one rule operation is performed in the second rule. As discussed above, Calvignac only discloses testing of a key to determine which rules shall be applied.

Based on the above distinctions, Applicant respectfully submits that the rejection of claim 45 under 35 U.S.C. § 102(e) is improper.

Rejection of Claim 52

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 52. Applicant respectfully traverses the

rejection because claim 52 includes limitations not found in any of the cited references. In its entirety, claim 52 reads as follows:

52. A method for providing access control in a computing system environment, the method comprising the steps of:

receiving an access request;

selecting, based on the access request, a set of rules containing multiple rules from at least one master set of rules, at least one of the multiple rules including multiple rule operations to be performed in sequential order;

for a first rule of the multiple rules:

performing a filter operation associated with the first rule to identify whether to execute any rule operations in the first rule; and

performing multiple operations in the first rule to determine whether to provide access to a storage system in response to receiving the access request, the first rule including a disregard instruction that, when executed, limits performance to fewer than all rule operations in a second rule of the selected set of rules as specified by disregard criteria in the disregard instruction.

The Examiner cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 52. The Examiner rejects claim 52 for the same reason as claim 45. Applicant respectfully submits that claim 52 includes limitations not found in claims 1, 19, and 45 and that the Examiner fails to provide an appropriate basis for the rejection of 52.

For example, claim 52 recites "at least one of the multiple rules including multiple rule operations to be performed in sequential order." The Examiner contends that FIGS. 3C-3D disclose this claim limitation in claim 52. Applicant respectfully disagrees with this assertion. As discussed above, there is no

indication in Calvignac that a respective rule includes multiple rule operations that are executed in sequential order. The cited passages and figures only disclose that rules are selected or discarded to produce subsets of rules.

Calvignac discloses testing a key against the rules and utilizes a tree for making a routing decision. However, Calvignac does not disclose that the subset of rules that match the key rules (or rules identified at leaf nodes of the decision tree) each include multiple rule operations. Nor does Calvignac disclose, teach, or suggest execution of a rule operation in which one of the selected rules, the rule operation being a disregard instruction that, when executed, causes fewer than all rule operations in another rule to be executed.

See Calvignac at column 13, line 30 to column 14 line 6. The Examiner's comparison of Calvignac to this cited passage to reject the claimed invention does not make sense. For example, this cited passage recites a technique of selecting rules for a particular dimension of a header packet. The rules are not executed until after a final decision of which rules to apply. The Examiner makes an improper rejection based on mistakenly interpreting that performance of the rules occurs during a selection process. There is no indication that the selected rules (e.g., those at leaf nodes of the decision tree) in Calvignac include a disregard instruction that limits performance of other rule operations. Each filter rule has a corresponding range of values that is used to create the decision tree that is used to select applicable rules at leaf nodes of the tree. Thus, the cited passages do not disclose every claim limitation. Accordingly, Applicant respectfully requests allowance of claim 52 over the cited art.

Rejection of Claim 57

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 57. Applicants respectfully traverse the

rejection because claim 57 includes limitations not found in any of the cited references. In its entirety, claim 57 reads as follows:

57. A method as in claim 53, wherein performance of the IF-THEN operation includes identifying whether an application generating the access request uses a particular resource in the storage system as well as whether a requestor associated with the access request is a member of a particular specified group and, if so, performing the rule operations in the first rule.

Applicant points out that claim 57 (which depends on claim 53, which depends on claim 52) recites "wherein performance of the IF-THEN operation includes identifying whether an application generating the access request uses a particular resource in the storage system as well as whether a requestor associated with the access request is a member of a particular specified group and, if so, performing the rule operations in the first rule."

The Examiner cites a portion of Calvignac discussing how so-called filter rules associated with data packets are used to decide what action to take with the data packet. A key is constructed from bits in a respective header of a data packet. There is no indication that the header information in Calvignac identifies whether an application generating a respective data packet uses a particular resource in a respective storage system. Also, there is no indication that the header information identifies whether a requestor associated with the access request is a member of a particular specified group. Accordingly, the claim includes limitations not cited by Calvignac and the rejection of claim 57 is improper. Applicant respectfully requests allowance of claim 57.

Rejection of Claim 58

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 58. Applicants respectfully traverse the

rejection because claim 58 includes limitations not found in any of the cited references. In its entirety, claim 58 reads as follows:

58. A method for providing access control in a computing system environment, the method comprising:

receiving an access request;

in response to receiving the access request, selecting a set of rules for processing to determine whether to permit the access request;

during processing of the set of rules, performing a conditional disregard rule operation in the set of rules;

based on performing the conditional disregard rule operation, disregarding execution of at least one rule operation other than the conditional disregard rule operation in the set of rules as specified by the conditional disregard rule operation; and

after performing the conditional disregard rule operation, performing at least one other rule operation in the set of rules not specified by disregard criteria in the conditional disregard rule operation.

The Examiner has rejected claim 58 for the same reasons he rejected claims 1, 45, and 52. Applicant respectfully submits that claim 58 includes limitations not found in these other claims and that the rejection is improper because claim 58 includes limitations not taught or suggested by Calvignac. As discussed above, Calvignac states: "Once the rules which the key may match are isolated, the key is explicitly tested against these rules" (Calvignac column 6, lines 34-37). In other words, Calvignac discloses a process of selecting rules based on the key and then applying the selected rules to the key. There is no indication that, after selection of the rules, the second stage of the two-stage process discussed above: "performs a conditional disregard rule operation in the set of rules; based on performing the conditional disregard rule operation, disregarding execution of at least one rule operation other than the conditional

disregard rule operation in the set of rules as specified by the conditional disregard rule operation; and after performing the conditional disregard rule operation, performing at least one other rule operation in the set of rules not specified by disregard criteria in the conditional disregard rule operation."

The claimed invention thus enables execution of one rule to affect certain selected rules while not affecting other selected rules. In other words, as indicated in claim 58, execution of a conditional disregard instruction causes non-execution of at least one other rule operation in a set of rules while at least one rule operation other than the non-executed rule operation is executed. Calvignac does not teach or suggest this claim limitation.

Rejection of Claim 60

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 60. Applicants respectfully traverse the rejection because claim 60 includes limitations not found in any of the cited references. In its entirety, claim 60 depends from claim 58 and reads as follows:

60. A method as in claim 58, wherein a field of data in the conditional disregard rule operation specifically identifies a first type of rule operations that are to be disregarded from execution in the set of rules, execution of the conditional disregard rule operation not having any affect on whether to perform a second type of rule operations in the set of rules.

Applicant submits that claim 60 further includes distinctions over Calvignac. For example, claim 60 recites "wherein a field of data in the conditional disregard rule operation specifically identifies a first type of rule operations that are to be disregarded from execution in the set of rules, execution of the conditional disregard rule operation not having any affect on whether to perform a second type of rule operations in the set of rules."

The Examiner cites column 8, lines 24-47 as teaching these limitations. Applicant submits that Calvignac does not disclose a disregard rule operation identifying a type of rule operation in a selected set of rules to be disregarded. Instead, this passage in Calvignac discloses sweeping through ranges to create subsets of rules that may be eventually applied to a key for purposes of making a forwarding decision. The teachings in Calvignac, therefore, are not equivalent to limitations in claim 60. Applicant respectfully submits that the rejection of claim 60 should be withdrawn.

Rejection of Claim 62

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 62. Applicants respectfully traverse the rejection because claim 62 includes limitations not found in any of the cited references. Claim 62 depends from claim 58 and reads as follows:

62. A method as in claim 58 further comprising:

during processing of the set of rules, performing an unconditional disregard rule operation in the set of rules that results in termination of performing any other rule operations in the selected set of rules.

Applicant disagrees with the Examiner's assessment that column 2, lines 40-45 of Calvignac teaches the limitations in claim 62. This passage recites matching a key of a packet to criteria in a filter rule. If there is a match, then a respective rule is applied to the packet. If not, a respective rule is not invoked and not applied to a packet. As discussed in Calvignac, this process is repeated to determine whether other rules apply to a key. There is no indication in Calvignac that the process of checking for matches stops after identifying a match.

As further cited by the Examiner at column 5 lines 28-35, Calvignac only indicates that ranges or exact matches can be used in the check process whether a rule applies to a key. Again, there is no indication that the process of checking stops for all other selected rules in a subset when there is no match. The rejection of claim 62 therefore should be withdrawn.

Rejection of Claim 63

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 63. Applicants respectfully traverse the rejection because claim 63 includes limitations not found in any of the cited references. Claim 63 reads as follows:

63. A method for providing access control in a computing system environment, the method comprising:

receiving an access request;

in response to receiving the access request, selecting a first set of rules and a second set of rules for processing to determine whether to permit the access request, the first set of rules and the second set of rules each including multiple rule operations;

during processing of the first set of rules, performing a disregard rule operation in the first set of rules; and

based on performing the disregard rule operation, disregarding execution of at least one rule operation in the second set of rules as identified by the disregard rule operation.

Applicant respectfully submits that claim 63 includes limitations not taught or suggested by Calvignac. For example, claim 63 recites "during processing of the first set of rules, performing a disregard rule operation in the first set of rules; and based on performing the disregard rule operation, disregarding execution of at least one rule operation in the second set of rules as identified by the

disregard rule operation." As discussed above, Calvignac supports a two-stage process of first creating subsets of rules and thereafter performing a matching process to select a set of rules. There is no indication in Calvignac executing a disregard rule operation that causes non-execution of certain other rule operations as indicated by the disregard rule operation. In fact, Calvignac does not mention that rules in any of the subsets reference each other in any way whatsoever.

Applicant respectfully requests allowance of claim 63.

Rejection of Claim 65

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 65. Applicants respectfully traverse the rejection because claim 65 includes limitations not found in any of the cited references. Claim 65 reads as follows:

65. A method as in claim 63 further comprising:

after disregarding execution of at least one rule operation in the second set of rules as identified by the disregard rule operation in the first set of rules, performing at least one rule operation in the second set of rules not associated with the disregard rule operation.

As discussed above, Calvignac discloses a technique of utilizing a key to select a set of rules to apply to a packet. There is no discussion whatsoever indicating how execution of one rule causes non-performance of other rules. Thus, Applicant respectfully requests withdrawal of the rejection and allowance of claim 65.

Rejection of Claim 67

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 67. Applicants respectfully traverse the rejection because claim 67 includes limitations not found in any of the cited references. Claim 67 depends from claim 63 and reads as follows:

67. A method as in claim 63, wherein the disregard rule operation is a conditional disregard rule operation, a field of data in the conditional disregard rule operation specifically identifying a first type of rule operations that are to be disregarded from execution in the first set of rules and the second set of rules, execution of the conditional disregard rule not having any affect on whether to perform a second type of rule operation in the second set of rules.

Calvignac supports a two-stage process of first creating subsets of rules and thereafter performing a matching process to select a set of rules. There is no indication in Calvignac executing a disregard rule operation that causes non-execution of certain other rule operations as indicated by the disregard rule operation, especially one in which "a field of data in the conditional disregard rule operation specifically identifying a first type of rule operations that are to be disregarded from execution in the first set of rules and the second set of rules, execution of the conditional disregard rule not having any affect on whether to perform a second type of rule operation in the second set of rules." In fact, Calvignac does not mention that rules in any of the executable rules reference each other that execution of one rule can affect another rule in any way whatsoever. Thus, Applicant respectfully requests withdrawal of the rejection and allowance of claim 67.

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 70. Applicants respectfully traverse the rejection because claim 70 includes limitations not found in any of the cited

references as discussed above for claim 1 as well as other claims. Claims 71-75 depend from claim 70 and include further distinctions not found in the cited references and therefore should also be allowable.

Rejection of Claim 70

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 70. Applicants respectfully traverse the rejection because claim 70 includes limitations not found in any of the cited references. Claim 70 reads as follows:

70. (Previously Presented) A method for providing access control in a computing system environment, the method comprising:

receiving an access request to access data in the computing system environment;

comparing the access request to a master rule set, each rule in the master rule set including a filter and a corresponding set of rule operations to be performed pending evaluation of the filter condition; and

for each rule containing a filter operation that evaluates to indicate execution of rule operations of that rule, executing the rule operations of that rule;

during execution of rule operations of that rule, executing a first conditional disregard instruction that establishes a first set of pre-conditions that must be met in successive rules in the master rule set in order for those successive rules to be executed after the rule containing the first conditional disregard instruction has been executed; and

executing at least one successive rule in the master rule set for which the access request meets the filters of those successive rules, and for which the first set of pre-conditions established by executing the first conditional disregard instruction are also met.

As discussed, Calvignac does not discuss a process of executing rule operations, but instead addresses selecting rules to be applied to a key. There certainly is no recitation in Calvignac of establishing preconditions based on execution of a rule, especially preconditions that must be met in order for successive rules to be executed after executing a conditional disregard instruction. The claimed technique affords a novel way of enabling one executed rule from eliminating other rules or rule operations from also being applied to an access request. There is no such teaching in Calvignac or any other cited reference. The rejection of claim 70 is improper and therefore should be withdrawn.

Rejection of Claim 72

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 72. Applicants respectfully traverse the rejection because claim 72 includes limitations not found in any of the cited references. Claim 72 depends from claim 71 which depends from claim 70 and reads as follows:

72. The method of claim 71 wherein pre-conditions established by execution of the conditional disregard instructions indicate a type of data upon which rule operations of successive rules in the master rule set operate that are not to be executed during execution of the successive rules in the master rule set.

The cited passages in Calvignac do not discuss a process of establishing preconditions, especially those that indicate a type of data upon which successive rules are not to be executed in a selected set of rules. The claimed technique affords a novel way of indicating enabling the rules themselves to determine which type of rules shall be eliminated during an execution process.

There is no such teaching in Calvignac or any other cited reference. The rejection of claim 72 is improper and therefore should be withdrawn.

Rejection of Claim 76

The Office Action cites Calvignac (U.S. Patent 6,539,394) as the closest prior art to reject the invention as in claim 76. Applicants respectfully traverse the rejection because claim 76 includes limitations not found in any of the cited references. Claim 76 reads as follows:

76. A computer program product having a computer-readable medium including computer program logic encoded thereon that when executed on a computer system provides a method for controlling access to a resource, and wherein when the computer program logic is executed on a processor in the computer system, the computer program logic causes the processor to perform the operations of:

receiving an access request to access data in the computing system environment;

comparing the access request to a master rule set, each rule in the master rule set including a filter and a corresponding set of rule operations to be performed pending evaluation of the filter condition; and

for each rule containing a filter operation that evaluates to indicate execution of rule operations of that rule, executing the rule operations of that rule;

during execution of rule operations of that rule, executing a first conditional disregard instruction that establishes a first set of pre-conditions that must be met in successive rules in the master rule set in order for those successive rules to be executed after the rule containing the first conditional disregard instruction has been executed; and

executing at least one successive rule in the master rule set for which the access request meets the filters of those successive rules, and

for which the first set of pre-conditions established by executing the first conditional disregard instruction are also met.

As discussed, Calvignac does not discuss a process of executing rule operations, but instead addresses selecting rules to be applied to a key. There certainly is no recitation in Calvignac of executing a first conditional disregard instruction that establishes a first set of pre-conditions that must be met in successive rules in the master rule set in order for those successive rules to be executed after the rule containing the first conditional disregard instruction has been executed; and executing at least one successive rule in the master rule set for which the access request meets the filters of those successive rules, and for which the first set of pre-conditions established by executing the first conditional disregard instruction are also met.establishing preconditions based on execution of a rule, especially one indicating preconditions that must be met in order for successive rules to be executed after executing a conditional disregard instruction. The claimed technique affords a novel way of enabling one executed rule from eliminating other rules or rule operations from also being applied to an access request. There is no such teaching in Calvignac or any other cited reference. The rejection of claim 76 is improper and therefore should be withdrawn.

(viii) Claims Appendix

Attached below.

(ix) Evidence Appendix

None

(x) Related Proceedings Appendix

None

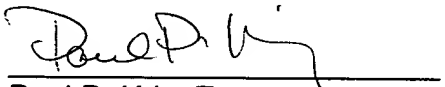
CONCLUSION

In view of the foregoing remarks, Applicants submit that the pending claims are in condition for allowance and request that the application be passed to issue. If the Examiner would like to discuss any of the pending claims, the Examiner is encouraged to call the Applicant(s) Representative at the number below.

Applicants hereby petition for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-3735.

If the enclosed papers or fees are considered incomplete, the Patent Office is respectfully requested to contact the undersigned Attorney at (508) 616-9660, in Westborough, Massachusetts.

Respectfully submitted,



Paul P. Kriz, Esq.
Attorney for Applicant(s)
Registration No.: 45,752
Chapin Intellectual Property Law, LLC
Westborough Office Park
1700 West Park Drive
Westborough, Massachusetts 01581
Telephone: (508) 616-9660
Facsimile: (508) 616-9661

Attorney Docket No.: EMC00-03(00011)

Dated: March 3, 2006

APPENDIX OF PENDING CLAIMS PRIOR TO MAILING OF FINAL OFFICE ACTION

1. (Previously Presented) A method for providing access control in a computing system environment, the method comprising the steps of:

receiving an access request;

selecting, based on the access request, a set of rules containing at least one rule from a master set of rules; and

producing an access control decision based on performing rule operations in a given rule of the selected set of rules by sequentially performing rule operations in the given rule until performing a disregard instruction, the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing; and

after performing the disregard instruction in the given rule:

i) evaluating the disregard criteria against any remaining unperformed rule operations in other rules of the selected set of rules, the other rules being rules other than the given rule;

ii) marking any remaining unperformed rule operations in the other rules of the selected set of rules that match the disregard criteria to be disregarded from further rule processing; and

iii) executing remaining unmarked rule operations in the other rules in the selected set of rules.

2. (Cancelled)

3. (Cancelled)

4. (Cancelled)

5. (Cancelled)

6. (Previously Presented) The method of claim 1 wherein the selected set of rules is arranged hierarchically such that rules containing rule operations that are more specific are performed before rule operations that are more general.

7. (Canceled)

8. (Canceled)

9. (Previously Presented) The method of claim 1 wherein the step of selecting includes the steps of:

determining an identity of a resource in the computing system environment to which access is requested in the access request; and

applying at least one filter operation, using the identity of the resource, for rules in the at least one master set of rules to produce the selected set of rules for use in determining the access control decision to the resource; and

wherein the method further includes the step of determining a role identity of a requestor submitting the access request; and

wherein the step of performing includes sequentially processing each rule operation in the selected set of rules using the role identity of the requestor submitting the access request in combination with the identity of the resource to determine if the requestor using the role identity can access the resource.

10. (Canceled)

11. (Canceled)

12. (Previously Presented) The method of claim 1 wherein:

the selected set of rules is arranged hierarchically such that rules containing rule operations that are more specific are performed before rules containing rule operations that are more general such that placement of the

disregard instruction in one of the rules in the selected set of rules causes the step of performing to control an amount of access control provided to a requestor that submitted the access request for access to a respective resource.

13. (Previously Presented) The method of claim 1 wherein the disregard instruction is a conditional instruction that has a condition that must be met before the disregard instruction is performed.

14. (Original) The method of claim 1 wherein:

at least one rule in the selected set of rules contains a relation that defines a condition based on a group definition; and

wherein at least one of the steps of selecting and performing includes the step of:

performing the relation to determine if at least one of a requestor, an access, and a resource specified in the access request satisfy the condition based on the group definition.

15. (Canceled)

16. (Canceled)

17. (Canceled)

18. (Canceled)

19. (Previously Presented) A computer system configured to provide access control, the computer system comprising:

at least one input/output interface;

a processor;

a memory system encoded with an authorization program;

at least one authorization database;

an interconnection mechanism coupling the processor, the at least one input/output interface, the memory system, and the at least one authorization database;

based at least in part on the processor executing the authorization program, the processor supporting steps of:

receiving an access request;

selecting, based on the access request, a set of rules containing at least one rule from a master set of rules;

producing an access control decision based on performing rule operations in a given rule of the selected set of rules by sequentially performing rule operations in the given rule until performing a disregard instruction, the disregard instruction including disregard criteria identifying a type of other rule operations in the selected set of rules to disregard from performing; and

after performing the unconditional disregard instruction in the given rule:

i) evaluating the disregard criteria against any remaining unperformed rule operations in other rules of the selected set of rules, the other rules being rules other than the given rule;

ii) marking any remaining unperformed rule operations in the other rules of the selected set of rules that match the disregard criteria to be disregarded from further rule processing; and

iii) executing remaining unmarked rule operations in the other rules in the selected set of rules.

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (Cancelled)

24. (Previously Presented) The computer system of claim 19 wherein the selected set of rules is arranged hierarchically such that when the processor performs the authorization program, rules containing rule operations that are more specific are performed before rule operations that are more general.

25. (Canceled)

26. (Canceled)

27. (Original) The computer system of claim 19 wherein when the processor performs the authorization program to select a selected set of rules, the processor:

determines an identity of an resource to which access is requested in the access request; and

applies at least one filter operation, using the identity of the resource, for rules in the at least one master set of rules to produce the selected set of rules for use in determining the access control decision to the resource; and

wherein when the processor performs the authorization program, the processor determines a role identity of a requestor submitting the access request; and

wherein the processor sequentially processes each rule operation in the selected set of rules using the role identity of the requestor submitting the access request in combination with the identity of the resource to determine if the requestor using the role identity can access the resource.

28. (Cancelled)

29. (Cancelled)

30. (Previously Presented) The computer system of claim 19 wherein:

the selected set of rules is arranged hierarchically such that rules containing rule operations that are more specific are performed by the processor before rules containing rule operations that are more general such that placement of the disregard instruction in one of the at least one rules in the selected set of rules causes the authorization program, when performed on the processor, to control an amount of access control provided to the requestor that submitted the access request for access to the resource.

31. (Previously Presented) The computer system of claim 27 wherein the disregard instruction is a conditional instruction that has a condition that must be met during processing by the processor before the disregard instruction is performed.

32. (Original) The computer system of claim 19 wherein:

at least one rule in the selected set of rules contains a relation that defines a condition based on a group definition; and

wherein when the processor performs at least one of the operations of selecting and performing, the processor performing the relation to determine if at least one of a requestor, an access, and a resource specified in the access request satisfy the condition based on the group definition.

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Previously Presented) A method for controlling applicability of rule operations in a rule-based access control system, the method comprising the step of:

selecting at least two rules for performance to determine an access control decision, the at least two rules including a first rule and a second rule;

performing a rule operation in the first rule of the at least two rules, the rule operation including a disregard instruction that, when performed, causes non-performance of at least one rule operation in the second rule that is disregarded based on the disregard instruction; and

performing at least one rule operation in the second rule other than the at least one rule operation in the second rule that is disregarded.

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Previously Presented) A method for providing access control in a computing system environment, the method comprising the steps of:

receiving an access request;

selecting, based on the access request, a set of rules containing multiple rules from at least one master set of rules, at least one of the multiple rules including multiple rule operations to be performed in sequential order;

for a first rule of the multiple rules:

performing a filter operation associated with the first rule to identify whether to execute any rule operations in the first rule; and

performing multiple operations in the first rule to determine whether to provide access to a storage system in response to receiving the access request, the first rule including a disregard instruction that, when executed, limits performance to fewer than all rule operations in a second rule of the selected set of rules as specified by disregard criteria in the disregard instruction.

53. (Previously Presented) A method as in claim 52, wherein the filter operation is an IF-THEN operation and performance of the IF-THEN operation provides an indication whether to perform rule operations in the first rule.

54. (Canceled)

55. (Previously Presented) A method as in claim 52, wherein the disregard instruction is a conditional disregard instruction, which limits a performance of other rule operations in multiple rules other than the first rule in the selected set of rules depending on occurrence of a corresponding condition as specified by the disregard criteria in the disregard instruction.

56. (Previously Presented) A method as in claim 55 further comprising:

performing at least one other rule operation in the first rule as well as other rules in the selected set of rules after performing the conditional disregard instruction.

57. (Previously Presented) A method as in claim 53, wherein performance of the IF-THEN operation includes identifying whether an application generating the access request uses a particular resource in the storage system as well as whether a requestor associated with the access request is a member of a particular specified group and, if so, performing the rule operations in the first rule.

58. (Previously Presented) A method for providing access control in a computing system environment, the method comprising:

receiving an access request;

in response to receiving the access request, selecting a set of rules for processing to determine whether to permit the access request;

during processing of the set of rules, performing a conditional disregard rule operation in the set of rules;

based on performing the conditional disregard rule operation, disregarding execution of at least one rule operation other than the conditional disregard rule

operation in the set of rules as specified by the conditional disregard rule operation; and

after performing the conditional disregard rule operation, performing at least one other rule operation in the set of rules not specified by disregard criteria in the conditional disregard rule operation.

59. (Previously Presented) A method as in claim 58 further comprising:

comparing disregard criteria in a data field associated with the conditional disregard rule operation to data in other rule operations to identify which other rule operations in the selected set of rules to disregard from performance.

60. (Previously Presented) A method as in claim 58, wherein a field of data in the conditional disregard rule operation specifically identifies a first type of rule operations that are to be disregarded from execution in the set of rules, execution of the conditional disregard rule operation not having any affect on whether to perform a second type of rule operations in the set of rules.

61. (Previously Presented) A method as in claim 60, wherein performing a conditional disregard rule operation further comprises identifying disregard criteria in the conditional disregard rule operation, the method further comprising:

upon performing the conditional disregard rule operation, marking any remaining unperformed rule operations in the set of rules as identified by the disregard criteria; and

continuing performance of rule operations in the set of rules that are not marked to be disregarded.

62. (Previously Presented) A method as in claim 58 further comprising:

during processing of the set of rules, performing an unconditional disregard rule operation in the set of rules that results in termination of performing any other rule operations in the selected set of rules.

63. (Previously Presented) A method for providing access control in a computing system environment, the method comprising:

receiving an access request;

in response to receiving the access request, selecting a first set of rules and a second set of rules for processing to determine whether to permit the access request, the first set of rules and the second set of rules each including multiple rule operations;

during processing of the first set of rules, performing a disregard rule operation in the first set of rules; and

based on performing the disregard rule operation, disregarding execution of at least one rule operation in the second set of rules as identified by the disregard rule operation.

64. (Previously Presented) A method as in claim 63, wherein selecting the first set of rules and the second set of rules includes applying a respective first filter and a second filter to identify whether to select the first set of rules and the second set of rules for execution.

65. (Previously Presented) A method as in claim 63 further comprising:

after disregarding execution of at least one rule operation in the second set of rules as identified by the disregard rule operation in the first set of rules, performing at least one rule operation in the second set of rules not associated with the disregard rule operation.

66. (Previously Presented) A method as in claim 63 further comprising:

following completion of executing the first set of rules and the second set of rules, generating an access control decision whether to permit the access request.

67. (Previously Presented) A method as in claim 63, wherein the disregard rule operation is a conditional disregard rule operation, a field of data in the conditional disregard rule operation specifically identifying a first type of rule operations that are to be disregarded from execution in the first set of rules and the second set of rules, execution of the conditional disregard rule not having any affect on whether to perform a second type of rule operation in the second set of rules.

68. (Previously Presented) A method as in claim 67, wherein performing a conditional disregard rule operation includes identifying disregard criteria in the conditional disregard rule operation, the method further comprising:

upon performing the conditional disregard rule operation, marking any remaining unperformed rule operations in the first set of rules and the second set of rules as identified by the disregard criteria; and

continuing performance of rule operations in the first set of rules and the second set of rules that are not marked to be disregarded.

69. (Previously Presented) A method as in claim 67 further comprising:

during processing of the first set of rules, performing an unconditional disregard rule operation that results in termination of performing all other rule operations in the selected first set of rules and the second set of rules.

70. (Previously Presented) A method for providing access control in a computing system environment, the method comprising:

receiving an access request to access data in the computing system environment;

comparing the access request to a master rule set, each rule in the master rule set including a filter and a corresponding set of rule operations to be performed pending evaluation of the filter condition; and

for each rule containing a filter operation that evaluates to indicate execution of rule operations of that rule, executing the rule operations of that rule;

during execution of rule operations of that rule, executing a first conditional disregard instruction that establishes a first set of pre-conditions that must be met in successive rules in the master rule set in order for those successive rules to be executed after the rule containing the first conditional disregard instruction has been executed; and

executing at least one successive rule in the master rule set for which the access request meets the filters of those successive rules, and for which the first set of pre-conditions established by executing the first conditional disregard instruction are also met.

71. (Previously Presented) The method of claim 70 wherein executing only the successive rules in the master rule set comprises:

executing a second conditional disregard instruction that establish a second set of pre-conditions that must also be met in addition to the first set of pre-conditions established by the first disregard instruction for any remaining successive rules in the master rule set to be executed.

72. (Previously Presented) The method of claim 71 wherein pre-conditions established by execution of the conditional disregard instructions indicate a type of data upon which rule operations of successive rules in the master rule set operate that are not to be executed during execution of the successive rules in the master rule set.

73. (Previously Presented) The method of claim 72 wherein the filter of at least one rule in the master rule set includes a test of whether an application associated with the access request uses a particular resource associated with the request.

74. (Previously Presented) The method of claim 72 wherein the filter of at least one rule in the master rule set includes a test of whether at least two resources associated with the access request are connected to each other.

75. (Previously Presented) The method of claim 72 comprising skipping execution of those successive rules in the master rule set for which the access request does not meet the filters of those successive rules, and for which the first and second set of pre-conditions established by executing the first and second disregard instructions are not met.

76. (Previously Presented) A computer program product having a computer-readable medium including computer program logic encoded thereon that when executed on a computer system provides a method for controlling access to a resource, and wherein when the computer program logic is executed on a processor in the computer system, the computer program logic causes the processor to perform the operations of:

receiving an access request to access data in the computing system environment;

comparing the access request to a master rule set, each rule in the master rule set including a filter and a corresponding set of rule operations to be performed pending evaluation of the filter condition; and

for each rule containing a filter operation that evaluates to indicate execution of rule operations of that rule, executing the rule operations of that rule;

during execution of rule operations of that rule, executing a first conditional disregard instruction that establishes a first set of pre-conditions that must be met in successive rules in the master rule set in order for those successive rules to be executed after the rule containing the first conditional disregard instruction has been executed; and

executing at least one successive rule in the master rule set for which the access request meets the filters of those successive rules, and for which the first

set of pre-conditions established by executing the first conditional disregard instruction are also met.

77. (Previously Presented) A method as in claim 45, wherein the rule-based access control system enables access to a storage resource in a storage area network, the method further comprising:

prior to selecting the first rule and the second rule, executing a respective filter operation associated with the first rule to identify whether to select the first rule and execute rule operations in the first rule;

prior to selecting the second rule and after selecting the first rule, executing rule operations in the first rule including the disregard instruction;

prior to selecting the second rule and after executing the first rule, executing a respective filter operation associated with the second rule to identify whether to select the second rule and execute rule operations in the second rule; and

after selecting and executing the first rule and after selecting the second rule, executing rule operations in the second rule as well as disregarding execution of at least one rule operation in the second rule based on execution of the disregard instruction in the first rule.

78. (Previously Presented) A method as in claim 77, wherein performing the rule operation in the first rule includes performing a conditional disregard instruction that identifies a particular type of rule operation to disregard from execution in the selected at least two rules, the method further comprising:

disregarding execution of a rule operation of the particular type in the second rule.

79. (Previously Presented) A method as in claim 78 further comprising:

performing a rule operation in the second rule that results in termination of a process of sequentially testing whether additional rules apply to the access request.

80. (Previously Presented) A method as in claim 79 further comprising:

selectively executing rule operations associated with the first rule and the second rule depending on: i) a type of data associated with the access request, ii) an amount of space available associated with the storage resource, and iii) a membership class of a user generating the access request.

81. (Previously Presented) A method as in claim 52, wherein the computing system environment enables access to a storage resource in a storage area network, the method further comprising:

prior to selecting the first rule and the second rule, executing a respective filter operation associated with the first rule to identify whether to select the first rule and execute rule operations in the first rule;

after selecting the first rule, executing rule operations in the first rule including the disregard instruction that limits execution of other rule operations;

prior to selecting the second rule and after executing the first rule, executing a respective filter operation associated with the second rule to identify whether to select the second rule and execute rule operations in the second rule; and

after selecting and executing the first rule and after selecting the second rule, executing rule operations in the second rule as well as disregarding execution of at least one rule operation in the second rule based on execution of the disregard instruction in the first rule.

82. (Previously Presented) A method as in claim 81 further comprising:

selectively executing rule operations associated with the first rule and the second rule depending on: i) a type of data associated with the access request, ii) an amount of space available associated with the storage resource, and iii) a membership class of a user generating the access request.